Statistical learning

Anadeal - Academic year 2024-2025

Introduction

A latent variable is a variable which is not directly observable (its realized value is hidden) **and** it is assumed to affect the observed variables.

Remark: other denomination: 'random effects, 'common factors', 'latent classes', 'underlying variables', 'frailties'

Latent variables are used in statistical models for:

- modeling unobserved heterogeneity in the data between units (latent variables are used to represent the effect of these unobservable factors)
- for measurement errors (the latent variables represent the "true" outcomes and the observed variables represent their "disturbed" versions)

The properties of latent variables must be inferred indirectly using a statistical model connecting them to the observations.

Introduction

Observed Variables \ Latent Variables	Continuous	Categorical
continuous	common factor model; SEM; Mixed models	latent profile model
categorical	latent trait model	latent class model

Some well-known latent variable models

- *Factor analysis model:* fundamental tool in multivariate statistics to summarize several (continuous) measurements through a small number of (continuous) latent traits.

- *Item Response Theory models:* models for items (categorical responses) measuring a common latent trait assumed to be continuous (or less often discrete) and typically representing an ability or a psychological attitude; the most important IRT model was proposed by Rasch (1961).

- *Generalized Linear mixed models:* extension of the class of linear models for continuous or categorical responses which account for unobserved heterogeneity.

- *Finite mixture model:* model, used even for a single response variable, in which subjects are assumed to come from subpopulations having different distributions of the response variables.

- *Latent Markov models:* models for longitudinal data in which the response variables are assumed to depend on an unobservable Markov chain.

Finite Mixture model

A real data motivation: Newcomb data

- The american astronomer and mathematician Simon Newcomb built in 1882 an experiment which primary goal is to determine the speed of light and to quantify the uncertainty of the measurement.
- Newcomb measured the time it took for light to travel from his lab to a mirror placed on the Washington Monument and back to his lab. The distance of the path traveled is 7.44373 km.

Data Set The given values are measures of the travel time $T = (24800 + Y)$ in nanoseconds.

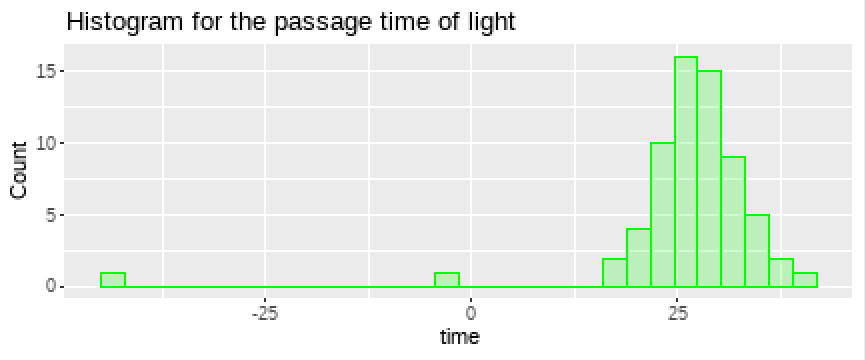
https://www2.stat.duke.edu/courses/Spring99/sta110a/newcomb.html

Ref: http://vigo.ime.unicamp.br/~fismat/newcomb.pdf

A real data motivation: Newcomb data

```
ggplot(data=Newcomb, aes(time)) +
  geom_histogram( col="green",
                  fill="green",
                  alpha = .2) +
  labs(title="Histogram for the passage time of light") +
  labs(x="time", y="Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Dealing with outliers

Once the outliers are identified you may consider one of the following approaches.

1. *Imputation* Imputation with mean / median / mode.
2. *Capping* For missing values that lie outside the 1.5 * IQR limits, we could cap it by replacing those observations outside the lower limit with the value of 5th percentile and those that lie above the upper limit, with the value of 95th percentile.
3. *Prediction* In yet another approach, the outliers can be replaced with missing values (NA) and then can be predicted by considering them as a response variable. We already discussed how to predict missing values.

Modeling the presence of outliers

$\mathcal{Y} = (Y_1, \ldots, Y_n)$ iid from a two components mixture distribution between a normal $\mathcal{N}(\mu, \sigma^2)$ and a uniform $\mathcal{U}(-a, a)$.

The uniform distribution is here to model the outliers in the data. The proportion of outliers being $(1 - \pi)$.

The pdf of the mixture is written as follows:

$$f_Y(y|\theta) = \pi\phi(y; \mu, \sigma^2) + (1 - \pi)c,$$

where $\theta = (\pi, \mu, \sigma^2)^t$ and $c = (2a)^{-1}$, $\phi(.\,)$ is the normal pdf.

Remark: we want to estimate θ, particularly μ. The parameters π and σ^2 are called *nuisance parameters*.

Mixture models

Mixture model:

$$f(x) = \sum_{j=1}^{k} \pi_j f_j(x), \ \sum_{j=1}^{k} \pi_j = 1, \ \pi_j > 0.$$

The weights are associated to a missing data structure (latent, unobserved), other parameters are related to the observation.

Mixture serves as useful extension of standard distributions.

Goal of inference in mixture model:

- clustering,
- provide estimation of parameters withing the different groups,
- estimate the number of groups.

Observed-data likelihood function

Statistical model: $\left\{P_\theta; \theta \in \Theta \subset [0,1] \times \mathbb{R} \times \mathbb{R}^+\right\}$

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} f(y_i; \theta) \\ &= \prod_{i=1}^{n} \left[\pi\phi(y_i; \mu, \sigma^2) + (1-\pi)c\right]. \end{aligned}$$

Difficult to maximize !

Expectation-Maximization algorithm

In practice, it is generally hard to find MLE ! $\longrightarrow$ numerical solutions. Many class of algorithms. Here we study the *Expectation Maximization* algorithm.

Two main applications for the EM algorithm:

- missing data
- analytically intractable likelihood functions

conditions of use: when the likelihood can be simplified by introducing *latent variables*.

Introducing a latent variable

Let introduce a *latent variable*

$$Z_i = \begin{cases} 1, & \text{if } Y_i \text{ is not an outlier} \\ 0, \text{else.} \end{cases}$$

The vector $Z = (Z_1, \ldots, Z_n)$ is unobserved and $Z_i \sim Be(\pi)$

- Y: observed data
- $X = (Y, Z)$: *complete data*
- Assume the joint density exists: $p(x;\theta) = p(y, z;\theta) = p(y|z;\theta)p(z;\theta)$.

Likelihood functions

Complete-data likelihood

$$L_c(\theta) = \prod_{i=1}^{n} f(y_i, z_i; \theta) = \prod_{i=1}^{n} f(y_i | z_i; \mu, \sigma^2) f(z_i; \pi)$$
$$= \prod_{i=1}^{n} \left[\phi(y_i; \mu, \sigma^2)^{z_i} c^{1-z_i} \pi^{z_i} (1-\pi)^{1-z_i} \right].$$

Complete-data log-likelihood

$$l_c(\theta) = \sum_{i=1}^{n} z_i \log \phi(y_i; \mu, \sigma^2) + (n - \sum_{i=1}^{n} z_i) \log c$$
$$+ \sum_{i=1}^{n} z_i \log(\pi) + \sum_{i=1}^{n} (1 - z_i) \log(1 - \pi).$$

Remarks:

- Would be easy to maximize if we observe the Z_i's but we do not.
- This function is a random variable since the missing info is unknown, random.

Intuition for building an estimation procedure

idea

- imagine if we had the values of θ, then given our observations Y, we could predict the values of the Z's. The best predictor in a mean square sense is a conditional expectation... Best predictor of Z given Y is the function which minimizes

$$\mathbb{E}\left[Z|Y\right] = \arg\min_g \mathbb{E}(Z - g(Y))^2.$$

- Once we have predicted the Z's, we can estimate the parameters considering observing the complete data.
- Then iterate and hope for convergence...

EM algorithm

Set $t = 0$, initialize $\theta^{(0)}$. Then,

1: given $\theta^{(t)}$ and the observed data Y formulate the conditional density $p(z|y, \theta^{(t)})$ for the complete data

2: using $p(z|y, \theta^{(t)})$, form the conditional expected complete log-likelihood denoted as $Q(\theta, \theta^{(t)})$

$$Q(\theta, \theta^{(t)}) = \mathbb{E}_{\theta^{(t)}} \left[l_c(\theta) | Y \right].$$

3: find $\theta^{(t+1)}$ that maximizes $Q(\theta, \theta^{(t)})$ in θ

5: set $t = t + 1$ and go to step 2 until reach some convergence criteria

$$\left\| \theta^{(t+1)} - \theta^{(t)} \right\| < \epsilon, \ \epsilon > 0.$$

$$|l(\theta^{(t+1)}) - l(\theta^{(t)})| < \epsilon, \ \epsilon > 0.$$

Back the the example: the Q function

Since $l_c(\theta)$ is linear in the Z_i's, the Q function is:

$$Q(\theta, \theta^{(t)}) = \sum_{i=1}^{n} z_i^{(t)} \log \phi(y_i; \mu, \sigma^2) +$$

$$\left(n - \sum_{i=1}^{n} z_i^{(t)}\right) \log c + \sum_{i=1}^{n} \left(z_i^{(t)} \log \pi + (1 - z_i^{(t)}) \log(1 - \pi)\right).$$

with $z^{(t)} \;= \mathbb{E}_{\theta^{(t)}}\left[Z_i | y_i\right] = \frac{\phi(y_i; \mu^{(t)}, \sigma^{2,(t)}) \pi^{(t)}}{\phi(y_i; \mu^{(t)}, \sigma^{2,(t)}) \pi^{(t)} + c(1 - \pi^{(t)})}$

$$\pi^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} z_i^{(t)},$$

$$\mu^{(t+1)} = \frac{\sum_{i=1}^{n} z_i^{(t)} y_i}{\sum_{i=1}^{n} z_i^{(t)}},$$

$$\sigma^{(t+1),2} = \frac{\sum_{i=1}^{n} z_i^{(t)} \left(y_i - \mu^{(t+1)}\right)^2}{\sum_{i=1}^{n} z_i^{(t)}}.$$

code EM-step

Description R code Python code

Perform one step of the EM algorithm.

Parameters:

```
  Y: data array
  p: parameter vector `\((\pi, \mu, \sigma)\)`
  c: constant (denisty of uniform dist.)
```

Returns:

```
  Dictionary with updated parameters and latent variable z
```

code EM-step

Description R code Python code

```r
emstep = function(Y,p,c)
{
  # p: vector of parameters c(pi,mu,sigma)
  N = length(Y)
  z = rep(0, length = N)

  # E-step
  phi = sapply(Y,dnorm,mean=p[2],sd=p[3])
  z = phi*p[1]/(phi*p[1]+c*(1-p[1]))

  # M-step
  S =  sum(z)
  p[1] = S/N
  p[2] = sum(z*Y) / S
  p[3] = sqrt(sum(z*(Y-p[2])^2) / S)

  return(list(p = p, z = z))
}
```

code EM-step

Description R code Python code

```python
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

def emstep(Y, p, c):
    N = len(Y)
    z = np.zeros(N)

    # E-step
    phi = norm.pdf(Y, loc=p[1], scale=p[2])
    z = (phi * p[0]) / (phi * p[0] + c * (1 - p[0]))

    # M-step
    S = np.sum(z)
    p[0] = S / N
    p[1] = np.sum(z * Y) / S
    p[2] = np.sqrt(np.sum(z * (Y - p[1])**2) / S)

    return {"p": p, "z": z}
```

code: iteration till convergence criterion met

Description

R code

Python code

Perform robust density estimation using the EM algorithm.

Parameters:

```
  Y: data array
  p: initial parameter vector `\((\pi, \mu, \sigma)\)`
  c: constant
  epsilon: convergence criterion
```

Returns:

```
  Dictionary with estimated parameters and latent variable z
```

code: iteration till convergence criterion met

Description R code Python code

```r
robust_dens = function(Y,p,c,epsilon)
{
  eps= 1
  p_i = p
  while (eps>epsilon)
  {
    pb = emstep(Y,p_i,c)
    eps = sum(abs(p_i-pb$p))
    p_i = pb$p
  }
  p_pred = pb$p
  z_pred = pb$z
  return(list(p_pred = p_pred, z_pred = z_pred))
}
```

code: iteration till convergence criterion met

Description R code **Python code**

```python
def robust_dens(Y, p, c, epsilon):
    eps = 1
    p_i = np.array(p)

    while eps > epsilon:
        result = emstep(Y, p_i, c)
        eps = np.sum(np.abs(p_i - result["p"]))
        p_i = result["p"]

    return {"p_pred": result["p"], "z_pred": result["z"]}
```

Numerical application

EM estimates

(Soft)-clustering

```
p_inits = c(0.5, 30,10)
est = robust_dens(Y = Newcomb$time, p = p_inits, c = 1/40, epsilon = 0.
round(est$p_pred,2)
```

```
## [1]  0.88 27.68  4.56
```

```
cat("Estimated speed of light",  10^9*(7.44373 / (27.68+24800)),"km/s",
```

```
## Estimated speed of light 299815.8 km/s
```

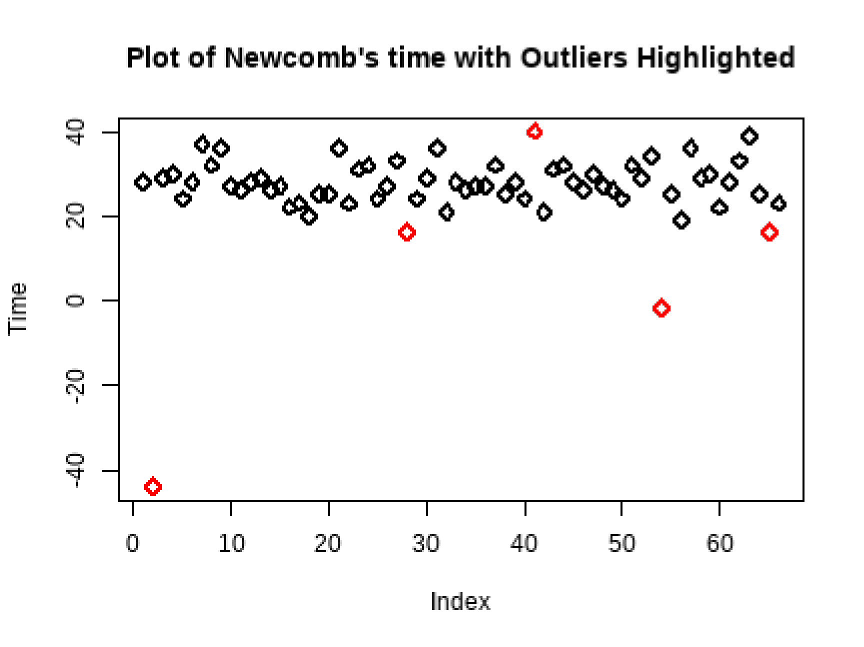
Numerical application

EM estimates

(Soft)-clustering

```
# Identifying outliers based on the condition (est$z_pred > 0.5)
Newcomb$time[est$z_pred <= 0.5]
```

```
## [1] -44  16  40  -2  16
```



Convergence of the EM algorithm

Lemma

At each step of the EM algorithm $l(\theta^{(t+1)}) \geq l(\theta^{(t)})$, always hold. **Monotonicity**

Furthermore, if $\theta^{(t)} \to \hat{\theta}$ and for every iteration $\frac{\partial Q(\theta_1,\theta_2)}{\partial \theta_2}|_{(\theta_1,\theta_2)=(\theta^{(t)},\theta^{(t+1)})} = 0$, then $\frac{\partial l_n(\theta)}{\partial \theta}|_{\theta=\hat{\theta}} = 0$ This point being a saddle point, a local or the global maximum.

Remark: one can show that this algorithm will converge to a point $\hat{\theta}$ where $\frac{\partial l_n(\theta)}{\partial \theta}|_{\theta=\hat{\theta}} = 0$.

There is no general convergence theorem

Speed of convergence to a stable point

To be distinguished from the rate of convergence of an estimator to the true value !

The rate at which $\theta^{(t)}$ converges to a stable point $\hat{\theta}$ is measured by the *ratio of the current iteration to the previous iteration*. Under certain regularity conditions it is only *linear* for the EM algorithm, i.e., $\exists T > 0$ and $0 < C < 1$, such that:

$$\left\|\theta^{(t+1)} - \hat{\theta}\right\| \leq C \left\|\theta^{(t)} - \hat{\theta}\right\|, \text{ for all } t \geq T.$$

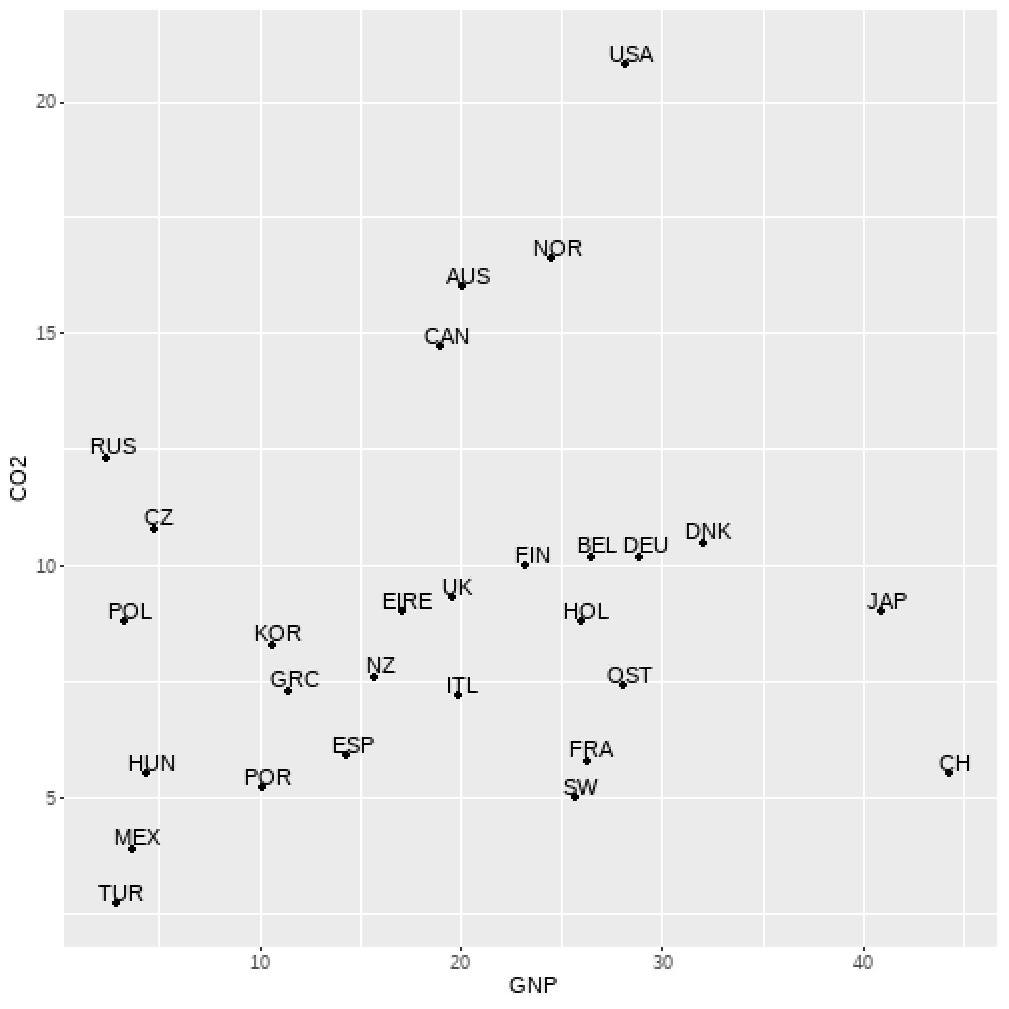
Mixture of regressions

Example taken from the *R-package mixtools*. The 'CO2data' data set gives the gross national product (GNP) per capita in 1996 for various countries as well as their estimated carbon dioxide (CO2) emission per capita for the same year. It consists in a data frame of 28 countries and the following columns:

- GNP: The gross national product per capita in 1996.
- CO2: The estimated carbon dioxide emission per capita in 1996.
- country: An abbreviation pertaining to the country measured (e.g., "GRC" = Greece and "CH" = Switzerland).

References: Hurn, M., Justel, A. and Robert, C. P. (2003) Estimating Mixtures of Regressions, Journal of Computational and Graphical Statistics 12(1), 55-79.

Mixture of regressions



Mixture of regressions

$$Y = \begin{cases} X\beta_1 + \epsilon_1, \ \epsilon_1 \sim \mathcal{N}(0, \sigma_1^2) \text{ if } Z = 1 \\ \vdots \\ X\beta_K + \epsilon_K, \ \epsilon_K \sim \mathcal{N}(0, \sigma_K^2) \text{ if } Z = K \end{cases}$$

Let $\theta = (\beta_1, \ldots, \beta_K, \sigma_1^2, \ldots, \sigma_K^2, \pi_1, \ldots, \pi_K)$

$$f_Y(y) = \sum_{k=1}^{K} \pi_k \phi(y; x\beta_k, \sigma_k^2).$$

The observed likelihood

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} f_Y(y_i; x_i, \theta) \\ &= \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \phi(y; x_i\beta_k, \sigma_k^2). \end{aligned}$$

Mixture of regressions

Complete data likelihood (introducing the latent variables Z)

$$\begin{aligned} L_c(\theta) &= \prod_{i=1}^{n} f(y_i, z_i; x_i, \theta) \\ &= \prod_{i=1}^{n} f(y_i | z_i; x_i, \theta) p(z_i | \pi) \\ &= \prod_{i=1}^{n} \prod_{k=1}^{K} \phi(y_i; x_i \beta_k, \sigma_k^2)^{z_{ik}} \pi_k^{z_{ik}}. \end{aligned}$$

where

$$z_{ik} = \begin{cases} 1, \text{ if } z_i = k, \\ 0, \text{ else.} \end{cases}$$

Mixture of regressions

$$l_c(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log(y_i; x_i\beta_k, \sigma_k^2) + \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log\pi_k.$$

$$Q(\theta, \theta^{(t)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}^{(t)}\log(y_i; x_i\beta_k, \sigma_k^2) + \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}^{(t)}\log\pi_k.$$

where

$$z_{ik}^{(t)} = \mathbb{E}_{\theta^{(t)}}\left[Z_{ik}|y_i\right] = P_{\theta^{(t)}}\left[Z_{ik} = k|y_i\right] = \frac{\phi\left(y_i; x_i\beta_k^{(t)}, \sigma_k^{2,(t)}\right)\pi_k^{(t)}}{\sum_{l=1}^{K}\phi\left(y_i; x_i\beta_l^{(t)}, \sigma_l^{2,(t)}\right)\pi_l^{(t)}}.$$

Mixture of regressions

Maximizing $Q(\theta, \theta^{(t)})$ gives:

$$\pi_k^{(t+1)} = \frac{N_k^{(t)}}{N}, \; N_k = \sum_{i=1}^{n} z_{ik}^{(t)}.$$

The terms depending on β_k

$$SS_k = \sum_{i=1}^{n} z_{ik}^{(t)} (y_i - x\beta_k)^2.$$

Mixture of regressions

This is a weighted least square minimization problem

$$SS_k = (y - X\beta)' W_k (y - X\beta_k).$$

$$W_k = Diag(z_{1k}, \ldots, z_{nk}).$$

$$\beta_k^{(t+1)} = (X' W_k X)^{-1} X' W_k Y.$$

and for the variance

$$\sigma_k^{2,(t+1)} = \frac{1}{N_k^{(t)}} (y - X\beta_k^{(t+1)})' W_k (y - X\beta_k^{(t+1)}).$$

Mixture of regressions

```
attach(CO2data)
CO2reg <- regmixEM(CO2, GNP, lambda = c(1, 3) / 4,
                   beta = matrix(c(8, -1, 1, 1), 2, 2), sigma = c(2, 1)
```

```
## number of iterations= 10
```

```
summary(CO2reg)
```

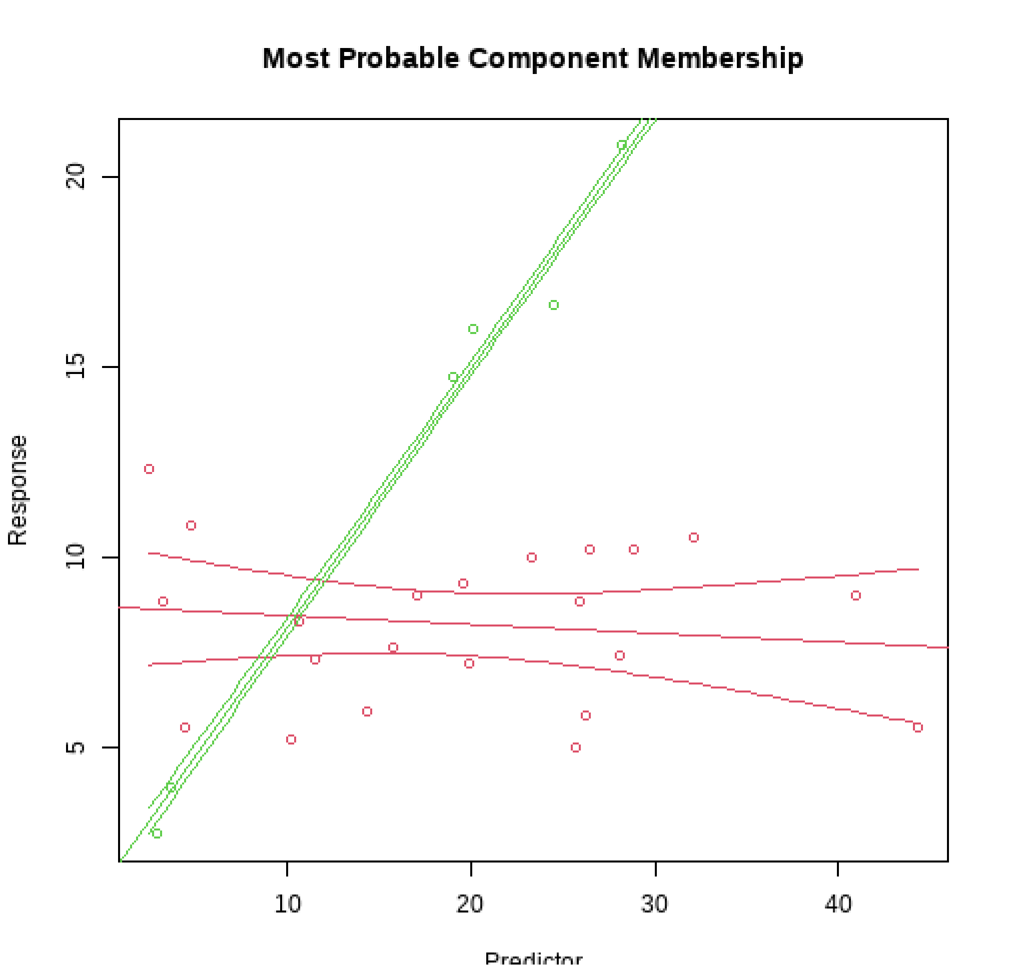
```
## summary of regmixEM object:
##           comp 1   comp 2
## lambda  0.754921 0.245079
## sigma   2.049315 0.809389
## beta1   8.678987 1.415150
## beta2  -0.023344 0.676596
## loglik at estimate:  -66.93977
```

```
clust = apply(CO2reg$posterior, 1, which.max)
CO2data$country[clust==1]
```

```
##  [1] JAP  KOR  NZ   OST  BEL  CZ   DNK  FIN  FRA  DEU  GRC  HUN  EIRE ITL
## [16] POL  POR  ESP  SW   CH   UK   RUS
## 28 Levels: AUS BEL CAN CH CZ DEU DNK EIRE ESP FIN FRA GRC HOL HUN ITL    U
```

Mixture of regressions

```
plot(CO2reg, density = TRUE, alpha = 0.01, whichplots = 2)
```


Most Probable Component Membership
Response
Predictor